UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/226,939	01/08/1999	JOHN K. VINCENT	346872000500	8916

23910 7590 01/17/2003

FLIESLER DUBB MEYER & LOVEJOY, LLP
FOUR EMBARCADERO CENTER
SUITE 400
SAN FRANCISCO, CA 94111

EXAMINER
LY, ANH

ART UNIT	PAPER NUMBER
2172	

DATE MAILED: 01/17/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 09/226,939	Applicant(s) VINCENT ET AL.	
	Examiner Anh Ly	Art Unit 2172	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *25 October 2002* .

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *9-29* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *9-29* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

Response to Arguments

1. Applicant's arguments filed on 10/25/2002 with respect to claims 9-29 have been considered but are moot in view of the new ground(s) of rejection.

2. Claims 1-8 and 30 are allowed on page #10, Office Action dated 08/07/2001.

3. Claims 9-29 are pending in this application.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was

not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

5. Claims 9-29 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent No. 5,325,531 issued to McKeeman et al. (hereinafter McKeeman) in view of US Patent No. 5,546,571 issued to Shan et al. (hereinafter Shan).

With respect to claim 9, McKeeman discloses dependency of code object and for each dependency found and all basic dependencies are generated into dependency tree (dependency analysis table from a makefile to gent dependency information to generate and store in the dependency graphs: col. 5, lines 18-67, col. 6, lines 1-12; also see col. 17, lines 20-67).

Mckeeman does not explicitly indicate, "querying a database catalog and the query recursive."

However, Shan discloses query database and recursive query of a database (col. 1, lines 25-39 and col. 4, lines 64-67 and col. 5, lines 1-3; also see col. 6, lines 30-38 and col. 9, lines 55-67).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Mckeeman with the teachings of Shan so as to have a dependency graphs or trees for a recursive query of a database. This combination would provide highly efficient evaluation of recursive queries without any requirement that the number of iterations be known in advance (Shan - col. 4, lines 64-67 and col. 5, lines 1-3). Also, this will provide a plurality of

hierarchical iteration levels for a database structure for recursively-derived data items (Shan - col. 5, lines 4-10) in the dependency analysis and recursive path analysis DBMS procedures environment.

With respect to claims 10-11, Mckeeman discloses debugging tools as debugger (see col. 5, lines 18-67 and col. 6, lines 1-12).

With respect to claim 12-16, Mckeeman discloses testing tool; invalid entries for a database, dependencies among database code object; and a dependency graph tools (col. 2, lines 32-44; col. 6, lines 25-45; col. 5, lines 18-67 and col. 6, lines 1-12; tools for software development: col. 6, lines 1-12).

With respect to claim 17, Mckeeman discloses dependencies of a code object and then for each dependency found; all basic dependencies are generated into a dependency graph; using a parser on each the code objects in the dependency graph to identify DML statements that "fire" triggers so as to identify dependencies on triggers (col. 5, lines 18-67, col. 6, lines 1-12; also see col. 17, lines 20-67; see fig. 7, item 70,, parser: col. 18, lines 40-52).

Mckeeman does not explicitly indicate, "using a recursive algorithm for querying a database catalog and the query recursive."

However, Shan discloses query database and recursive query of a database (col. 1, lines 25-39 and col. 4, lines 64-67 and col. 5, lines 1-3; also see col. 6, lines 30-38 and col. 9, lines 55-67).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Mckeeman with the

teachings of Shan so as to have a dependency graphs or trees for a recursive query of a database. This combination would provide highly efficient evaluation of recursive queries without any requirement that the number of iterations be known in advance (Shan - col. 4, lines 64-67 and col. 5, lines 1-3). Also, this will provide a plurality of hierarchical iteration levels for a database structure for recursively-derived data items (Shan - col. 5, lines 4-10) in the dependency analysis and recursive path analysis DBMS procedures environment.

With respect to claims 18-21, McKeeman discloses debugging tools as debugger (see col. 5, lines 18-67 and col. 6, lines 1-12); testing tool; invalid entries for a database, dependencies among database code object; and a dependency graph tools (col. 2, lines 32-44; col. 6, lines 25-45; col. 5, lines 18-67 and col. 6, lines 1-12; tools for software development: col. 6, lines 1-12).

With respect to claim 22, McKeeman discloses dependency information and outputs a direct dependency graph of a database code object, the "direct dependency graph" containing dependencies that do not involve dependencies on triggers and on implementations of object oriented code objects in the database (col. 17, lines 20-67).

Mckeeman does not explicitly indicate, "applying a recursive algorithm for querying a database catalog and the query recursive."

However, Shan discloses query database and recursive query of a database (col. 1, lines 25-39 and col. 4, lines 64-67 and col. 5, lines 1-3; also see col. 6, lines 30-38 and col. 9, lines 55-67).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Mckeeman with the teachings of Shan so as to have a dependency graphs or trees for a recursive query of a database. This combination would provide highly efficient evaluation of recursive queries without any requirement that the number of iterations be known in advance (col. 4, lines 64-67 and col. 5, lines 1-3). Also, this will provide a plurality of hierarchical iteration levels for a database structure for recursively-derived data items (col. 5, lines 4-10) in the dependency analysis and recursive path analysis DBMS procedures environment.

With respect to claims 23-26, Mckeeman discloses debugging tools as debugger (see col. 5, lines 18-67 and col. 6, lines 1-12); testing tool; invalid entries for a database, dependencies among database code object; and a dependency graph tools (col. 2, lines 32-44; col. 6, lines 25-45; col. 5, lines 18-67 and col. 6, lines 1-12; tools for software development: col. 6, lines 1-12).

With respect to claim 27, McKeeman disclose a digital computer (see fig. 3); the data including object oriented code objects, specifications of packages, implementations of packages, specifications of types, implementations of types and triggers; and a code mechanism for generating a dependency graph, the dependency graph being a data structure and having entries to contain representations of depending code objects, of packages, implementations of packages, specifications of types, implementations of types, triggers and dependencies of triggers which are relevant to the target data base code object (col. 5, lines 18-67, col. 6, lines 1-12, col. 17, lines 20-67).

McKeeman does not explicitly, indicate, "a database server couple to the computer."

However, Shan discloses a computerized database system (see abstract).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Mckeeman with the teachings of Shan so as to have a dependency graphs or trees for a recursive query of a database. This combination would provide highly efficient evaluation of recursive queries without any requirement that the number of iterations be known in advance (Shan - col. 4, lines 64-67 and col. 5, lines 1-3). Also, this will provide a plurality of hierarchical iteration levels for a database structure for recursively-derived data items (Shan - col. 5, lines 4-10) in the dependency analysis and recursive path analysis DBMS procedures environment.

With respect to claim 28, McKeeman discloses the data including representations of object oriented code objects, specifications of packages, implementations of packages, specifications of types, implementations of types and triggers; and using a recursive code mechanism for generating a dependency graph, the dependency graph being a data structure and having entries to contain representations of dependent code objects, specifications of packages, implementations of packages, specifications of types, implementations of types, triggers and dependencies of triggers which are relevant to the target data base code object (col. 5, lines 18-67, col. 6, lines 1-12, col. 17, lines 20-67).

McKeeman does not explicitly, indicate, "a database server couple to the computer; and applying a recursive algorithm for querying a database catalog and the query recursive."

However, Shan discloses a computerized database system (see abstract); and query database and recursive query of a database (col. 1, lines 25-39 and col. 4, lines 64-67 and col. 5, lines 1-3; also see col. 6, lines 30-38 and col. 9, lines 55-67).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Mckeeman with the teachings of Shan so as to have a dependency graphs or trees for a recursive query of a database. This combination would provide highly efficient evaluation of recursive queries without any requirement that the number of iterations be known in advance (Shan - col. 4, lines 64-67 and col. 5, lines 1-3). Also, this will provide a plurality of hierarchical iteration levels for a database structure for recursively-derived data items (Shan - col. 5, lines 4-10) in the dependency analysis and recursive path analysis DBMS procedures environment.

With respect to claim 29, McKeeman discloses generating a dependency graph of the target database code object the dependency graph being a data structure and having entries to contain representations of depending code objects, specifications of packages, implementations of packages, triggers and dependencies of triggers which are relevant to the target data base code object; and a program code mechanism for using the dependency graph to debug the target data base code object (col. 5, lines 18-

67, col. 6, lines 1-12, col. 17, lines 20-67; debugger: see fig 2, item 22: col. 5, lines 18-67 and col. 6, lines 1-12; also see col. 29, lines 12-32).

McKeeman does not explicitly, indicate, "a recursive algorithm for querying a database catalog and the query recursive."

However, Shan discloses a computerized database system (see abstract); and query database and recursive query of a database (col. 1, lines 25-39 and col. 4, lines 64-67 and col. 5, lines 1-3; also see col. 6, lines 30-38 and col. 9, lines 55-67).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Mckeeman with the teachings of Shan so as to have a dependency graphs or trees for a recursive query of a database. This combination would provide highly efficient evaluation of recursive queries without any requirement that the number of iterations be known in advance (Shan - col. 4, lines 64-67 and col. 5, lines 1-3). Also, this will provide a plurality of hierarchical iteration levels for a database structure for recursively-derived data items (Shan - col. 5, lines 4-10) in the dependency analysis and recursive path analysis DBMS procedures environment.

Contact Information

6. Any inquiry concerning this communication should be directed to Anh Ly whose telephone number is (703) 306-4527 via E-Mail: **ANH.LY@USPTO.GOV**. The examiner can be reached on Monday - Friday from 8:00 AM to 4:00 PM.

If attempts to reach the examiner are unsuccessful, see the examiner's supervisor, Kim Vu, can be reached on (703) 305-4393.

Any response to this action should be mailed to:

Commissioner of Patents and Trademarks

Washington, D.C. 20231

or faxed to: (703) 746-7238 (for After Final communications intended for entry)

or: (703) 746-7239 (for formal/Official communications intended for entry)

or: (703) 746-7240 (for informal or draft communications or Customer Service Center , please label "PROPOSED" or "DRAFT")

Hand-delivered responses should be brought to Crystal Park II, 2121 Crystal Drive, Arlington, VA, Fourth Floor (receptionist).

Inquiries of a general nature or relating to the status of this application should be directed to the Group receptionist whose telephone number is (703) 305-9600.

HOSAIN T. ALAM
PRIMARY EXAMINER

AL

Jan. 8th, 2003.